

# LAB 1 : Prometheus

## Introduction

[Prometheus](#) est un puissant système de surveillance open source qui collecte les métriques de vos services et les stocke dans une base de données chronologique. Il offre un modèle de données multidimensionnel, un langage de requête flexible et diverses possibilités de visualisation grâce à des outils comme [Grafana](#).

Par défaut, Prometheus exporte uniquement des métriques sur lui-même (par exemple le nombre de requêtes reçues, sa consommation de mémoire, etc.). Mais vous pouvez étendre considérablement Prometheus en installant des *exportateurs*, des programmes optionnels qui génèrent des métriques supplémentaires.

Les exportateurs - à la fois ceux officiels gérés par l'équipe Prometheus et ceux fournis par la communauté - fournissent des informations sur tout, de l'infrastructure, des bases de données et des serveurs Web aux systèmes de messagerie, aux API, etc.

Certains des choix les plus populaires incluent:

- [node\\_exporter](#) : Cela produit des métriques sur l'infrastructure, y compris l'utilisation actuelle du processeur, de la mémoire et du disque, ainsi que des statistiques d'E/S et du réseau, telles que le nombre d'octets lus sur un disque ou la charge moyenne d'un serveur.
- [blackbox\\_exporter](#) : Cela génère des métriques dérivées de protocoles de détection tels que HTTP et HTTPS pour déterminer la disponibilité des points de terminaison, le temps de réponse, etc.
- [mysqld\\_exporter](#) : Il rassemble des métriques liées à un serveur MySQL, telles que le nombre de requêtes exécutées, le temps de réponse moyen des requêtes et l'état de réplication du cluster.
- [rabbitmq\\_exporter](#) : Ceci génère des métriques sur le système de messagerie [RabbitMQ](#), y compris le nombre de messages publiés, le nombre de messages prêts à être livrés et la taille de tous les messages de la file d'attente.
- [nginx-vts-exporter](#) : Cela fournit des métriques sur un serveur Web Nginx utilisant le [module Nginx VTS](#), y compris le nombre de connexions ouvertes, le nombre de réponses envoyées (regroupées par codes de réponse) et la taille totale des demandes envoyées ou reçues en octets.

Vous pouvez trouver une liste plus complète des exportateurs officiels et communautaires sur [le site Web de Prometheus](#).

Dans ce cours, vous allez installer, configurer et sécuriser Prometheus et Node Exporter pour générer des métriques qui faciliteront la surveillance des performances de votre serveur.

## Étape 1 - Création d'utilisateurs de service

Pour des raisons de sécurité, nous commencerons par créer deux nouveaux comptes utilisateurs, **prometheus**. Nous utiliserons ces comptes tout au long du cours pour isoler la propriété des fichiers et répertoires principaux de Prometheus.

Créez les utilisateurs et utilisez les options `--no-create-home` et `--shell /bin/false` afin que ces utilisateurs ne puissent pas se connecter au serveur.

```
sudo useradd --no-create-home --shell /bin/false prometheus
sudo useradd --no-create-home --shell /bin/false node_exporter
```

Avant de télécharger les binaires Prometheus, créez les répertoires nécessaires pour stocker les fichiers et les données de Prometheus. En suivant les conventions Linux standard, nous allons créer un répertoire sous `/etc` pour les fichiers de configuration de Prometheus et un répertoire sous `/var/lib` pour ses données.

```
sudo mkdir /etc/prometheus
sudo mkdir /var/lib/prometheus
```

Maintenant, définissez l'utilisateur et le groupe des nouveaux répertoires à **prometheus**.

```
sudo chown prometheus:prometheus /etc/prometheus
sudo chown prometheus:prometheus /var/lib/prometheus
```

Avec nos utilisateurs et répertoires en place, nous pouvons maintenant télécharger Prometheus puis créer le fichier de configuration minimal pour exécuter Prometheus pour la première fois.

## Étape 2 - Téléchargement de Prometheus

Tout d'abord, téléchargez et décompressez la version stable actuelle de Prometheus dans votre répertoire personnel. Vous pouvez trouver les derniers binaires ainsi que leurs sommes de contrôle sur la [page de téléchargement de Prometheus](https://prometheus.io/download/) : <https://prometheus.io/download/>

```
cd ~
curl -LO https://github.com/prometheus/prometheus/releases/download/v2.24.1/prometheus-2.24.1.linux-amd64.tar.gz
```

Ensuite, utilisez la `sha256sum` commande pour générer une somme de contrôle du fichier téléchargé :

```
sha256sum prometheus-2.24.1.linux-amd64.tar.gz
```

Comparez la sortie de cette commande avec la somme de contrôle sur la page de téléchargement Prometheus pour vous assurer que votre fichier est à la fois authentique et non corrompu.

## Output

```
e12917b25b32980daee0e9cf879d9ec197e2893924bd1574604eb0f550034d46
prometheus-2.0.0.linux-amd64.tar.gz
```

Si les sommes de contrôle ne correspondent pas, supprimez le fichier téléchargé et répétez les étapes précédentes pour re-télécharger le fichier.

Maintenant, décompressez l'archive téléchargée.

```
tar xvf prometheus-2.24.1.linux-amd64.tar.gz
```

Cela créera un répertoire appelé prometheus-2.24.1.linux-amd64 contenant deux fichiers binaires ( prometheus et promtool) et deux répertoires consoles et des console\_libraries contenant les fichiers d'interface Web, une licence, une notice et plusieurs fichiers d'exemple.

Copiez les deux binaires dans le /usr/local/bin répertoire.

```
sudo cp prometheus-2.24.1.linux-amd64/prometheus /usr/local/bin/
sudo cp prometheus-2.24.1.linux-amd64/promtool /usr/local/bin/
```

Définissez la propriété de l'utilisateur et du groupe sur les binaires sur l' utilisateur **prometheus** créé à l'étape 1.

```
sudo chown prometheus:prometheus /usr/local/bin/prometheus
sudo chown prometheus:prometheus /usr/local/bin/promtool
```

Copiez les répertoires consoles et console\_libraries dans /etc/prometheus.

```
sudo cp -r prometheus-2.24.1.linux-amd64/consoles /etc/prometheus
sudo cp -r prometheus-2.24.1.linux-amd64/console_libraries /etc/prometheus
```

Définissez la propriété de l'utilisateur et du groupe sur les répertoires sur l' utilisateur **prometheus**. L'utilisation de l'indicateur -R garantit que la propriété est également définie sur les fichiers à l'intérieur du répertoire.

```
sudo chown -R prometheus:prometheus /etc/prometheus/consoles
sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries
```

Enfin, supprimez les fichiers restants de votre répertoire personnel car ils ne sont plus nécessaires.

```
rm -rf prometheus-2.24.1.linux-amd64.tar.gz prometheus-2.24.1.linux-amd64
```

Maintenant que Prometheus est installé, nous allons créer ses fichiers de configuration et de service en préparation de sa première exécution.

## Étape 3 - Configuration de Prometheus

Dans le répertoire `/etc/prometheus`, utilisez `nano` ou votre éditeur de texte préféré pour créer un fichier de configuration nommé `prometheus.yml`. Pour l'instant, ce fichier contiendra juste assez d'informations pour exécuter Prometheus pour la première fois.

```
sudo nano /etc/prometheus/prometheus.yml
```

**Attention:** le fichier de configuration de Prometheus utilise le [format YAML](#), qui interdit strictement les **tabulations** et nécessite **deux espaces pour l'indentation**. Prometheus ne démarre pas si le fichier de configuration n'est pas formaté correctement.

Dans les paramètres `global`, définissez l'intervalle par défaut pour les collectes (scrapes). Notez que Prometheus appliquera ces paramètres à chaque exportateur à moins que les paramètres propres d'un exportateur individuel ne remplacent les globaux.

Fichier de configuration Prometheus partie 1 - `/etc/prometheus/prometheus.yml`

```
global:
  scrape_interval: 15s
```

Cette valeur `scrape_interval` indique à Prometheus de collecter des métriques de ses exportateurs toutes les 15 secondes, ce qui est suffisamment long pour la plupart des exportateurs.

Maintenant, ajoutez Prometheus lui-même à la liste des exportateurs à supprimer avec la directive `scrape_configs` suivante :

Fichier de configuration Prometheus, partie 2 - `/etc/prometheus/prometheus.yml`

```
...
scrape_configs:
- job_name: 'prometheus'
  scrape_interval: 5s
  static_configs:
  - targets: ['localhost:9090']
```

Prometheus utilise le `job_name` pour étiqueter les exportateurs dans les requêtes et les graphiques, alors assurez-vous de choisir quelque chose de descriptif ici.

Et, comme Prometheus exporte des données importantes sur lui-même que vous pouvez utiliser pour surveiller les performances et le débogage, nous avons remplacé la directive globale `scrape_interval` de 15 secondes à 5 secondes pour des mises à jour plus fréquentes.

Enfin, Prometheus utilise les directives `static_configs` et `targets` pour déterminer où les exportateurs s'exécutent. Étant donné que cet exportateur particulier est en cours d'exécution sur le même serveur que Prometheus lui-même, on peut utiliser `localhost` au lieu d'une adresse IP ainsi que le port par défaut, 9090.

Votre fichier de configuration devrait maintenant ressembler à ceci:

Fichier de configuration Prometheus - `/etc/prometheus/prometheus.yml`

```
global:
  scrape_interval: 15s
scrape_configs:
- job_name: 'prometheus'
  scrape_interval: 5s
  static_configs:
  - targets: ['localhost:9090']
```

Enregistrez le fichier et quittez votre éditeur de texte.

Maintenant, définissez la propriété de l'utilisateur et du groupe sur le fichier de configuration sur l'utilisateur **prometheus** créé à l'étape 1.

```
sudo chown prometheus:prometheus /etc/prometheus/prometheus.yml
```

Une fois la configuration terminée, nous sommes prêts à tester Prometheus en l'exécutant pour la première fois.

## Étape 4 - Exécution de Prometheus

Démarrez Prometheus en tant qu'utilisateur **prometheus**, en fournissant le chemin d'accès au fichier de configuration et au répertoire de données.

```
sudo -u prometheus /usr/local/bin/prometheus \
--config.file /etc/prometheus/prometheus.yml \
--storage.tsdb.path /var/lib/prometheus/ \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries
```

La sortie contient des informations sur la progression du chargement de Prometheus, le fichier de configuration et les services associés. Il confirme également que Prometheus écoute sur le port 9090.

Output

```
level=info ts=2017-11-17T18:37:27.474530094Z caller=main.go:215
msg="Starting Prometheus" version="(version=2.0.0, branch=HEAD,
revision=0a74f98628a0463dddc90528220c94de5032d1a0)"

level=info ts=2017-11-17T18:37:27.474758404Z caller=main.go:216
build_context="(go=go1.9.2, user=root@615b82cb36b6,
date=20171108-07:11:59)"

level=info ts=2017-11-17T18:37:27.474883982Z caller=main.go:217
host_details="(Linux 4.4.0-98-generic #121-Ubuntu SMP Tue Oct 10
14:24:03 UTC 2017 x86_64 prometheus-update (none))"

level=info ts=2017-11-17T18:37:27.483661837Z caller=web.go:380
component=web msg="Start listening for connections" address=0.0.0.0:9090

level=info ts=2017-11-17T18:37:27.489730138Z caller=main.go:314
msg="Starting TSDB"

level=info ts=2017-11-17T18:37:27.516050288Z caller=targetmanager.go:71
component="target manager" msg="Starting target manager..."
```

```
level=info ts=2017-11-17T18:37:27.537629169Z caller=main.go:326
msg="TSDB started"

level=info ts=2017-11-17T18:37:27.537896721Z caller=main.go:394
msg="Loading configuration file" filename=/etc/prometheus/prometheus.yml

level=info ts=2017-11-17T18:37:27.53890004Z caller=main.go:371
msg="Server is ready to receive requests."
```

Si vous obtenez un message d'erreur, vérifiez que vous avez utilisé la syntaxe YAML dans votre fichier de configuration, puis suivez les instructions à l'écran pour résoudre le problème.

Maintenant, arrêtez Prometheus en appuyant sur CTRL+C, puis ouvrez un nouveau fichier de service systemd.

```
sudo nano /etc/systemd/system/prometheus.service
```

Le fichier de service systemd indique d'exécuter Prometheus en tant qu'utilisateur **prometheus**, avec le fichier de configuration situé dans le répertoire `/etc/prometheus/prometheus.yml` et de stocker ses données dans le répertoire `/var/lib/prometheus`. (vous pouvez en savoir plus sur [Présentation des unités Systemd et des fichiers d'unité](#).)

Recopiez le contenu suivant dans le fichier:

Fichier de service Prometheus - `/etc/systemd/system/prometheus.service`

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
--config.file /etc/prometheus/prometheus.yml \
--storage.tsdb.path /var/lib/prometheus/ \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

Enfin, enregistrez le fichier et fermez votre éditeur de texte.

Pour utiliser le service nouvellement créé, rechargez systemd.

```
sudo systemctl daemon-reload
```

Vous pouvez maintenant démarrer Prometheus à l'aide de la commande suivante:

```
sudo systemctl start prometheus
```

Pour vous assurer que Prometheus est en cours d'exécution, vérifiez l'état du service.

```
sudo systemctl status prometheus
```

La sortie vous indique l'état de Prometheus, l'identifiant principal du processus (PID), l'utilisation de la mémoire, etc.

Si l'état du service n'est pas active, suivez les instructions à l'écran et retracez les étapes précédentes pour résoudre le problème avant de poursuivre le cours.

## Output

```
• prometheus.service - Prometheus
  Loaded: loaded (/etc/systemd/system/prometheus.service; disabled; vendor preset: enabl
  Active: active (running) since Tue 2021-02-09 18:50:33 UTC; 8s ago
  Main PID: 2209 (prometheus)
  Tasks: 9
  Memory: 19.8M
  CPU: 115ms
  CGroup: /system.slice/prometheus.service
          \u2514\u25002209 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml
```

...

Lorsque vous êtes prêt à continuer, appuyez sur Q pour quitter la commande status.

Enfin, activez le service au démarrage.

```
sudo systemctl enable prometheus
```

Maintenant que Prometheus est opérationnel, nous pouvons installer un exportateur supplémentaire pour générer des métriques sur les ressources de notre serveur.